AFRL-IF-RS-TR-2005-380
In-House Final Technical Report
November 2005

COMMERCIAL CROSS-DOMAIN XML GUARD

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-380 has been reviewed and is approved for publication

APPROVED: /s/

JON S. JONES, Chief
Fusion Technology Branch

FOR THE DIRECTOR: /s/

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 2005	**Final In-House, Oct 04 – Sep 05**

4. TITLE AND SUBTITLE

COMMERCIAL CROSS-DOMAIN XML GUARD

5. FUNDING NUMBERS

C - In House
PE - 62702F
PR - 459E
TA - H5
WU - B1

6. AUTHOR(S)

Timothy Holzmann, Yat Fu, Stuart Hirshfield, Ryan Cunningham, Justin Randall

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/IFEB
525 Brooks Road
Rome NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFED
525 Brooks Road
Rome NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2005-380

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Timothy Holzmann, 2Lt, USAF/IFEB/(315) 330-7568 Timothy.Holzmann@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

This technical report summarizes the work performed and the results from an in-house study conducted at the Rome Research Site on using COTS products in cross-domain applications and problems. The study involved two leading edge appliances, resulting in several interesting discoveries. The significant results are that both appliances clearly outperformed our baseline desktop solution; one clearly had the best performance, and the other was significantly easier to configure. Study documents objectives, methods, assumptions, testbed architectures, and results analysis.

14. SUBJECT TERMS

Cross-Domain, Guard, XML, COTS, Firewall, Evaluation

15. NUMBER OF PAGES 25

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

The overall objective of this effort was to evaluate the current capabilities and areas for development of commercial XML-based (eXtensible Markup Language) web-security products. Our expectation was that this effort would justify a follow-on effort to augment the capabilities of future ISSE Guard and STARGuard appliances. This report describes both how we met our objective and the implications of our findings to future in-house efforts in support of cross-domain information access.

INTRODUCTION

BACKGROUND

This effort was motivated most generally by the potential utility of XML and web-services technologies to cross-domain solutions. Our specific interest was in the rapidly evolving commercial market for “XML appliances” that address both processing and security concerns. While these commercial solutions tend to reflect security needs typical of the commercial sector (and require additional research to meet unique government security needs), their processing capabilities, both in terms of XML processing speeds and web-services functionality, far surpass any current or developing GOTS products. Our overall goal, then, was to secure, test, and evaluate a number of appliances in terms of their applicability to current and future guarding efforts.

APPROACH

Our intent was to select and secure for in-house use a small number of commercial security appliances and to benchmark them in a common guard-like scenario. The overall goal was divided into four sub-tasks which served as vector guidelines for our progress through this effort. In the next section, we will list the four sub-tasks and our accomplishments under each one. Throughout the effort, we found a number of unexpected supplemental efforts that were necessary to complete these sub-tasks. These supplemental tasks and their results are detailed in Appendix A.

METHODS, ASSUMPTIONS AND PROCEDURES

TASK 1: Select Commercial Appliances and Build Proficiency

<u>Goals</u>
Search the market for commercially available web-services security appliances.; select one to three appliances based on the potential for meeting our needs, the price, the current certification using the Common Criteria Evaluation, and ease of use; purchase or obtain for evaluation a sample product; develop proficient knowledge to use the product.

<u>Accomplishments</u>
During the summer of 2004 two student interns worked under our supervision to develop a comprehensive market review of available XML appliances. This report served as the starting point for our selection process. We convened numerous meetings and discussions with a cross-section of vendors, many of whom came on-sight to provide demos of their appliances. Ultimately, we settled on three vendor/appliances (Sarvega’s “XML Guardian Gateway,” Forum System’s “Sentry,” and Tumbleweed Communications’ “MailGate Email Firewall”) and secured versions of each for in-house testing.

The Sarvega and Forum Systems products fit the description of general purpose XML appliances (that is, they provide both accelerated XML processing and web services security). They were chosen based upon a variety of factors, including their performance specifications, their support for security features, the usability of their interfaces, the availability of technical support, and their demonstrated acceptance in commercial markets. Also, both products are at some stage in the submission/evaluation process for Common Criteria Evaluation (CCE) certification. The Tumbleweed product is primarily a security device aimed specifically at e-mail processing. This product was chosen (somewhat belatedly) because it represented a growing market of email-specific tools that seem to apply to future guarding requirements.

As regards developing proficiency in the use of these tools, we participated in extensive training sessions (conducted by the vendors) for each appliance. Throughout the process, we reviewed and conducted all of the supplied tutorial materials, and interacted extensively with both the Sarvega and Forum System. They were very responsive to questions and comments. Particularly for the Sarvega, we have found and reported a number of minor defects with their appliance, such as the appliance could not handle an attribute of maxOccurs=”unbounded” for an element in a schema file; it could not handle a schema file that is referencing another schema file; and it could not have multiple policies to handle different types of XML files under one configuration. However, their engineering team was able to provide patches and fixes for these defects in a timely manner. Furthermore, our evaluation of the products has already influenced new releases and product development plans. Subsequent to formal training, we developed our own in-house tutorials for configuring and using the appliances in our particular lab environment (see appendices).

Once the machines were secured and configured as standalone appliances in our lab, we wanted to develop a simple test application that would demonstrate connectivity and communication with our machines. We chose to develop a simple cross-domain chat (Jabber) server and client (see “Supplemental Efforts,” below) that would offload the digital signing and verifying of messages to the appliances. Because signing and verifying many small messages requires intensive resources from a guard, the servers and the clients, it is a reasonable and realistic test case (in terms of both processing and security demands) for the use of an XML appliance for security processing. The figure below shows a top-level view of how an XML appliance fits into the overall architecture.

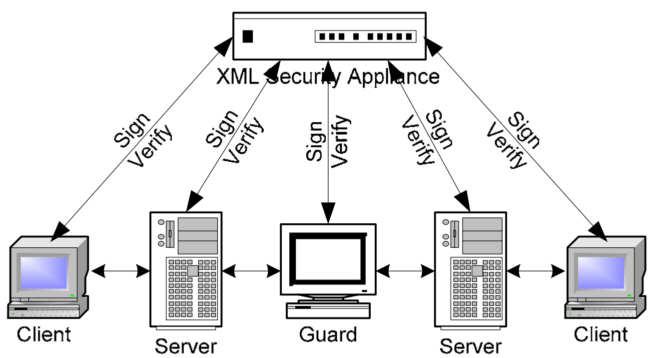


Figure 1: Jabber Use-Case Architecture

TASK 2: Develop Testbed

Goals

Develop a testbed to use the appliances to pass XML messages between two enclaves. In-house computers will be used to develop this testbed. The messages passed will be unclassified messages similar to those commonly used in common guarding environments. The testbed must produce a bandwidth requirement similar to current requirements of fielded systems, and it must be able to introduce a limited number of faulty data elements among the many requests that meet filter specifications.

Accomplishments

This task developed differently than was originally anticipated. We initially intended to compare the commercial appliances against fielded guard systems in situations similar to real world environments. However, subsequent studies demonstrated that configuring the guard systems to our lab environment would require efforts that were beyond the scope of this research project. Our research team maintained a flexible mindset to adjust to this new development, and an alternative testbed was developed. As we considered alternative methods of testing the appliances, we also availed ourselves of similar research that was recently completed (but not published) by the MITRE Corp. We researched their studies results and focused to provide additional research in some gaps that were left in their studies. In this new testbed we tested the appliances against themselves and against a standard desktop PC.

In testing against the desktop, we provide a reasonable baseline against which to compare the processing capabilities of these appliances. We developed a hand-coded Visual Basic program that uses a built-in XML program library to perform schema validation, digital signing, and encryption. This was not part of our original testing plan, and serves to show the dynamic nature of the XML market at large. At the time that we wrote our statement of work only a couple of programming languages supported XML processing, and did so very modestly (to the extent that XML documents could be read and parsed). In the time since, programming language developers have recognized the demand for more sophisticated XML processing, and have incorporated libraries and classes that can be easily accessed from within any program.

The complete testbed as developed is depicted in Figure 2 below. The desktop PC we used (called “Self” in our testbed interface) was a Pentium 4, 3.40 GHz IBM desktop with 1.0 GB of RAM and a Gigabit Network Interface Card. It is running under the Window XP with the Internet Information Services (IIS) version 5.1. The appliance that we received from Forum Systems is an evaluation version of the Forum Sentry (without a cryptographic acceleration card), and the one that we received from the Sarvega is the Sarvega Guardian. Additional specifications of the Forum Sentry and Sarvega Guardian cannot be specified in this document because they include proprietary vendor information. All of the machines are interconnected by a D-Link Gigabit switch, DGS-1008D. We configured the appliances to interface with our lab machines, and a software interface was developed to monitor the performance of the appliances (see Figure 3). This software interface allows the user to submit an arbitrarily large set of data files all contained in a single directory multiple times to each of the systems being tested. The time the system takes to process each time the directory is sent is recorded as well as the total number of bytes sent, the total length of time the test required, and the overall throughput.

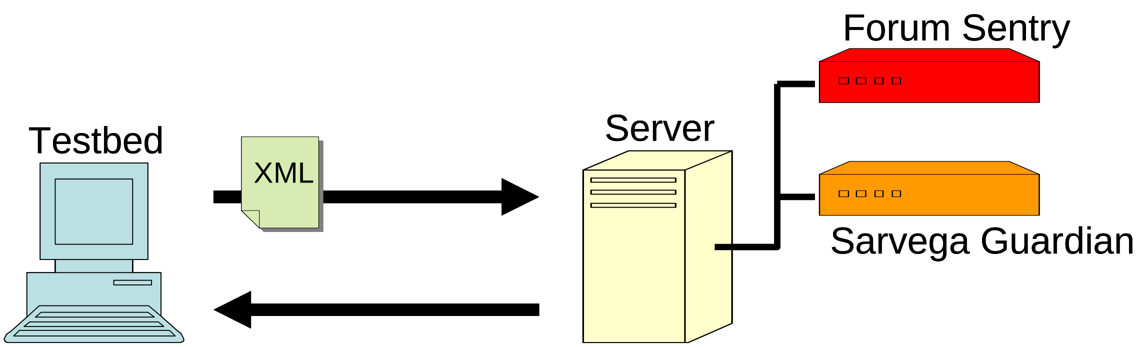


Figure 2: Testbed Architecture

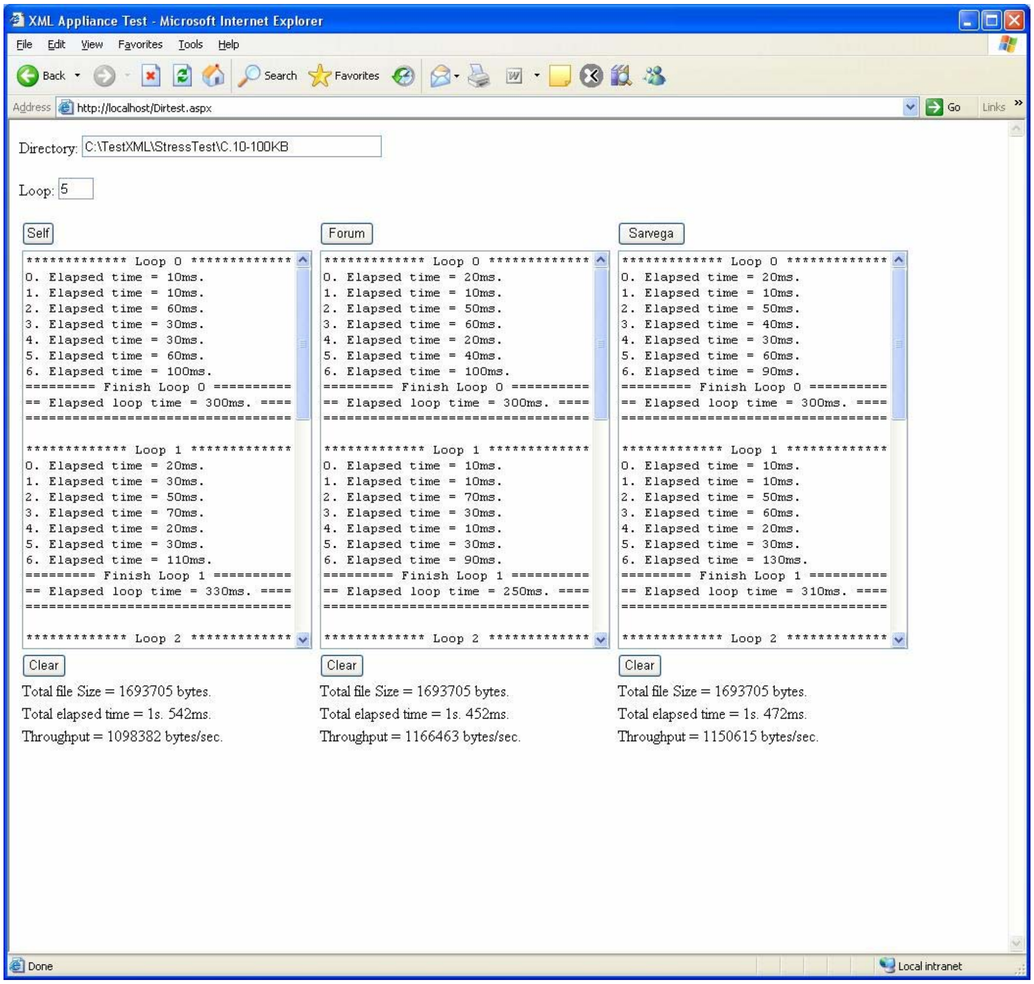


Figure 3: Testbed Interface

Our sample XML data files were generated using Altova XMLSpy (free version available at www.altova.com) from schemas which we found in military use cases (such as MTIX) and harvested from the internet. To these files, we applied a prototypical processing pipeline that represents (in fact, it extends the capabilities of) typical guarding applications. This pipeline dictates that a document be read in, validated against an XML schema, be digitally signed and encrypted.

TASK 3: Test Appliances and Document Results

<u>Goals</u>

Test the appliances using XML messages of varying size, complexity and processing requirements. The results should document numerical comparisons on appliance throughput capabilities.

<u>Accomplishments</u>

Each appliance's performance was measured on two processing tasks. The first, labeled "SV," was the mere schema validation of an XML document. For uniformity in the tests, all XML instances were validated using only their root elements. The second, labeled "SV-Enc-Sign," performed schema validation followed by encryption and digital signing of the documents.

After harvesting the test results data, we also performed statistical analyses of the data. We developed two calculated percentages to measure our results. First we measured the latency of the security processing incurred as a percent of the normal processing latency using the formula:

$$x = \frac{[time\_for\_SV \& Sign \& Enc] - [time\_for\_SV]}{[time\_for\_SV]} * 100\%$$

$$= \frac{[time\_for\_SV \& Sign \& Enc]}{[time\_for\_SV]} * 100\% - 100\%$$

In addition, we measured the throughput with security processing as a percent of the throughput using the following formula:

$$y = \frac{[throughput\_for\_SV \& Sign \& Enc]}{[throughput\_for\_SV]} * 100\%$$

The results of this study and analysis are recorded in the "Results" section below.

TASK 4: Final Report

Goals
Fully document work performed under this effort. Maintain meeting minutes, update the plan schedule, document reasons why certain commercial appliances were selected, provide a detailed description of the testbed, include documented test results, summarize capability benefits and outline follow-on research requirements for augmenting current guard solutions with the selected products.

Accomplishments
This report fulfills this goal.

RESULTS

Appendix A shows the results of our testing procedures. It includes three tables detailing the results of our testing (Figures 4-6). Each chart shows the timing and throughput results one appliance as applied to a variety of file sizes, number of files, and the number of times files were dispatched for processing. Note that our testing software is not able to provide accurate time measurement for the total elapsed time fewer than 15 milliseconds. Hence, the total elapsed time for tests fewer than 15 milliseconds are highlighted in yellow color, and are not included for analysis (these tests were included for display purposes in figures 7 and 11, but the analysis calculations did not include those test cases). Since our XML instances were generated from the schemas, the complexity of the instances is a function of the complexity required and allowed by the source schemas as well as the size of the XML instance. The tests are overall increasing in size and complexity, but that growth is not uniform. For example, test ID 8 has fewer number of files and fewer total bytes than test ID 7, but that single file is larger than any of those included in ID 7, so that XML instance represents a higher complexity level.

The first of these three tables displays the timing and throughput results from the tests run on the Forum Systems Sentry appliance. The second displays comparable results for the Sarvega appliance. The third table shows timing and throughput results for our hand-coded Visual Basic program running on the desktop PC (which was simultaneously providing the testbed interface) and serves as an effective baseline for comparative purposes. Additional tests using a configured accredited guarding system may yield other useful results, but the configuring of an accredited guard was beyond the cost and time scope of this effort.

The analysis of this data has been summarized in the Figures 7-11 in Appendix A which compare the three XML processing platforms across: total processing times, time spent on overhead (communication time between

components), throughput rate for basic XML processing, throughput rate for security-related tasks, and time/throughput analysis. The final graph measures the relative performance of the Sarvega and Forum appliances as compared to the desktop solution.

Finally, by considering the *x* and *y* values as random variables, we used the student T-distribution to perform analysis of expected means. Our results of this test are presented in Figures 12 and 13 also in Appendix A.

CONCLUSION

We believe that this effort has conclusively demonstrated the benefit to the government in using COTS technologies for optimized XML processing in cross-domain applications. Especially in the lower file size range, the throughput benefits are well demonstrated by these tests. Overall, the Sarvega appliance outperforms both the Forum appliance and our desktop solution across all processing and size tests. Informally, though, the user interface for the Sarvega appliance is not as sophisticated as that of the Forum appliance. Thus, as one might expect, there is a tradeoff to be made between pure performance and usability of the appliance. Additionally, there is a necessary cost associated with the acquisition of these appliances both in dollars and in manpower necessary to interface, configure, and support their implementation. Thus, it is imperative that a cost-benefit analysis be performed specific to anticipate the use case before the appliances should be used in prototype or live scenarios. Depending on the implementation and use-case requirements in terms of throughput, configurability, training times, cost, etc., any one of these appliances might be the best fit, or possibly even a combination of several.

It is clear to us - and to many others within DoD - that XML processing demands will continue to increase for the foreseeable future, and that COTS technology in the form of XML appliances will evolve to meet those demands. We have already seen significant increases in functionality of these appliances, dramatic increases in performance, and a range of specialized product types (Tumbleweed's e-mail appliance is one example) that was hard to imagine even one year ago.

RECOMMENDATIONS

There are a number of related follow-up activities that grow naturally from our findings, many of which will be detailed in a forthcoming proposal. Briefly, we intend to develop an in-house prototype "guard" that is based on one (or more) of these XML appliances. There are at least two ways (see Figures 4 and 5 below) in which such appliances can be incorporated into an existing guard, and our experience to date indicates the distinct possibility that these appliances (at least in a prototype, lab setting) can serve as functional guards themselves. We intend to investigate this possibility by configuring COTS appliances so as to address as nearly as possible the functional requirements of a proposed guarding system, and to use this COTS-based system as the guarding component of an instance of a fielded AFRL system (for example, the Collaborative Gateway).

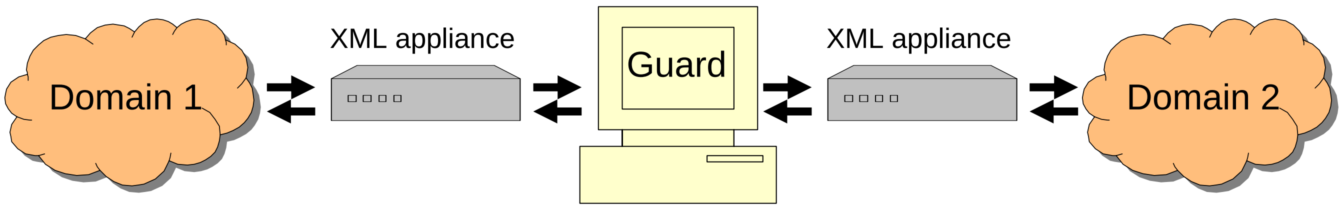


Figure 4: In-line configuration:
An appliance acts as a firewall/proxy/router for a particular domain.

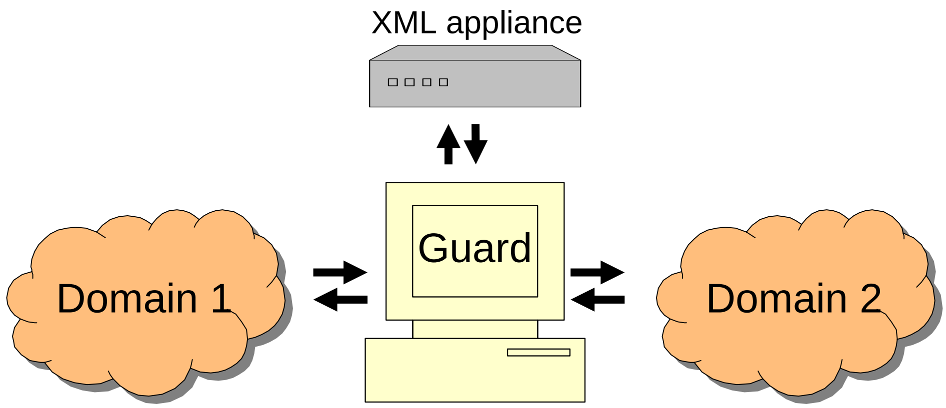


Figure 5: Side-car configuration:
An appliance handles all of the computational intensive XML processes for a Guard.

REFERENCES

A number of excellent resources are available to acquaint a user with XML. We found these especially useful:
 www.w3schools.com – Provides simple tutorials on the most common XML technologies
 www.w3.org – W3C website with links to all W3C recommendations regarding XML
We also provide the reader links to the appliance vendor's websites that were included in this study.
 www.forumsystems.com – Forum Systems' website
 www.sarvega.com – Sarvega's website
Finally this study made extensive use of Altova's XMLSpy software. This software is freely available at
 www.altova.com

Appendix A: Testing Results

ID	File size	# of files	# of loops	Total Size (B)	Forum Sentry *SV* Total elapsed time (s)	Throughput (B/s)	*SV - Enc - Sign* Total elapsed time (s)	Throughput (B/s)
1	200B - 5KB	16	1	25527			1.31	24759
2	200B - 5KB	16	2	51054			1.802	28332
3	200B - 5KB	16	3	76581			2.844	26927
4	200B - 5KB	16	5	127635			4.336	29436
5	200B - 5KB	16	10	255270			8.772	29101
6	200B - 5KB	16	15	382905			13.399	28577
7	200B - 5KB	16	20	510540			17.535	29115
8	5 - 10KB	6	1	44015			0.08	550188
9	5 - 10KB	6	2	88030			0.15	586867
10	5 - 10KB	6	3	132045			0.24	550188
11	5 - 10KB	6	5	220075			0.56	392991
12	5 - 10KB	6	10	440150			0.981	448675
13	5 - 10KB	6	15	660225			1.672	394871
14	5 - 10KB	6	20	880300			2.123	414649
15	10 - 100KB	7	1	338741	0.29	1168072	0.42	806526
16	10 - 100KB	7	2	677482	0.58	1168072	0.801	845795
17	10 - 100KB	7	3	1016223	0.821	1237787	1.523	663331
18	10 - 100KB	7	5	1693705	1.452	1166463	2.713	624292
19	10 - 100KB	7	10	3387410	2.084	1208063	4.917	688918
20	10 - 100KB	7	15	5081115	4.135	1228807	8.942	568230
21	10 - 100KB	7	20	6774820	5.547	1221348	12.157	557277
22	100KB - 1MB	15	1	4523159	3.274	1381539	5.728	789658
23	100KB - 1MB	15	2	9046318	6.028	1500716	10.194	887416
24	100KB - 1MB	15	3	1.4E+07	9.243	1468081	15.812	858176
25	100KB - 1MB	15	5	2.3E+07	15.392	1469321	26.337	858708
26	100KB - 1MB	15	10	4.5E+07	31.194	1450009	52.755	857390
27	1 - 2MB	5	1	7638636	4.957	1540980	9.193	830919
28	1 - 2MB	5	2	1.5E+07	10.535	1450144	18.496	825977
29	1 - 2MB	5	3	2.3E+07	15.572	1471610	26.938	850691
30	1 - 2MB	5	4	3.1E+07	20.749	1472579	36.903	827969
31	1 - 2MB	5	5	3.8E+07	26.47	1466318	44.343	861312
32	2 - 3MB	5	1	1.3E+07	8.962	1443400	15.652	826460
33	2 - 3MB	5	2	2.6E+07	18.736	1380844	31.915	810638
34	2 - 3MB	5	3	3.9E+07	27.699	1401034	48.429	801323
35	3 - 4MB	5	1	1.8E+07	14.35	1272596	23.43	792507
36	3 - 4MB	5	2	3.7E+07	26.377	1384672	44.544	819942
37	Over 4MB	6	1	3E+07	24.685	1218677	41.099	731965

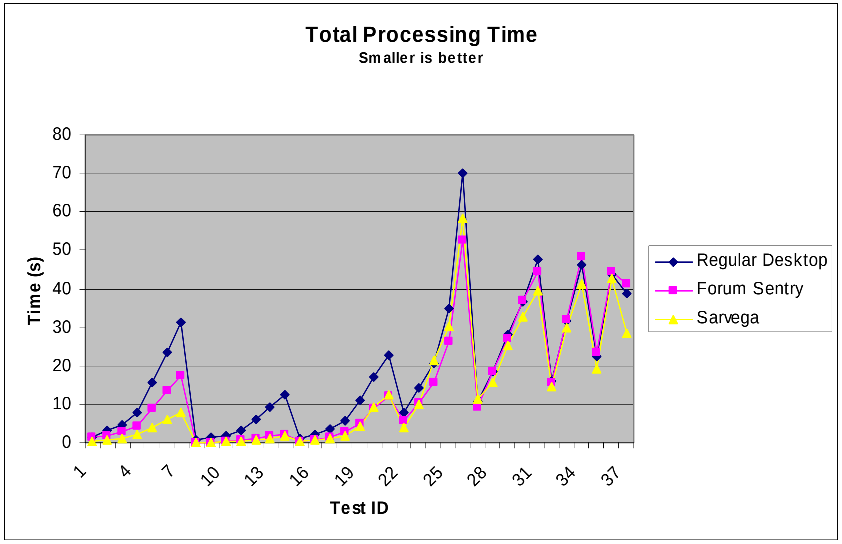
Figure 6: Test Results for the Forum Sentry appliance

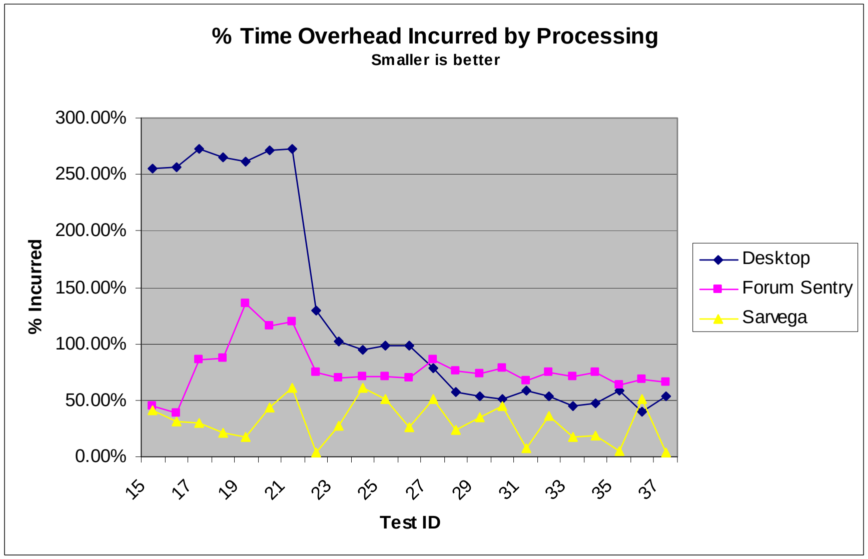
ID	File size	# of files	# of loops	Total Size (B)	Sarvega *SV* Total elapsed time (s)	Throughput (B/s)	*SV – Enc – Sign* Total elapsed time (s)	Throughput (B/s)
1	200B – 5KB	16	1	25527			0.41	62261
2	200B – 5KB	16	2	51054			0.811	62952
3	200B – 5KB	16	3	76581			1.201	63764
4	200B – 5KB	16	5	127635			2.012	63437
5	200B – 5KB	16	10	255270			3.975	64219
6	200B – 5KB	16	15	382905			5.938	64484
7	200B – 5KB	16	20	510540			7.911	64535
8	5 – 10KB	6	1	44015			0.07	628786
9	5 – 10KB	6	2	88030			0.15	586867
10	5 – 10KB	6	3	132045			0.25	528180
11	5 – 10KB	6	5	220075			0.43	511802
12	5 – 10KB	6	10	440150			0.821	536114
13	5 – 10KB	6	15	660225			1.231	536332
14	5 – 10KB	6	20	880300			1.682	523365
15	10 – 100KB	7	1	338741	0.27	1254596	0.38	891424
16	10 – 100KB	7	2	677482	0.62	1092713	0.811	835366
17	10 – 100KB	7	3	1016223	0.901	1127883	1.171	867825
18	10 – 100KB	7	5	1693705	1.472	1150615	1.782	950452
19	10 – 100KB	7	10	3387410	3.625	934458	4.256	795914
20	10 – 100KB	7	15	5081115	6.469	785456	9.283	547357
21	10 – 100KB	7	20	6774820	7.781	870688	12.497	542116
22	100KB – 1MB	15	1	4523159	3.835	1179442	3.985	1135046
23	100KB – 1MB	15	2	9046318	7.731	1170136	9.814	921777
24	100KB – 1MB	15	3	1.4E+07	13.128	1033629	21.19	640372
25	100KB – 1MB	15	5	2.3E+07	20.038	1128645	30.183	749289
26	100KB – 1MB	15	10	4.5E+07	46.496	972806	58.403	774474
27	1 – 2MB	5	1	7638636	7.661	997081	11.546	661583
28	1 – 2MB	5	2	1.5E+07	12.518	1220424	15.522	984233
29	1 – 2MB	5	3	2.3E+07	18.626	1230318	25.196	909506
30	1 – 2MB	5	4	3.1E+07	22.592	1352450	32.837	930491
31	1 – 2MB	5	5	3.8E+07	36.392	1049494	39.316	971441
32	2 – 3MB	5	1	1.3E+07	10.805	1197200	14.671	881722
33	2 – 3MB	5	2	2.6E+07	25.326	1021539	29.862	866369
34	2 – 3MB	5	3	3.9E+07	34.97	1109730	41.399	937396
35	3 – 4MB	5	1	1.8E+07	18.216	1002511	19.177	952279
36	3 – 4MB	5	2	3.7E+07	28.18	1296079	42.6	868366
37	1061879	####	####	1061879	1061879	1061879	1061879	1061879

Figure 7: Test Results for the Sarvega appliance

					Regular Desktop			
					SV		*SV – Enc – Sign*	
ID	File size	# of files	# of loops	Total Size (B)	Total elapsed time (s)	Throughput (B/s)	Total elapsed time (s)	Throughput (B/s)
1	200B – 5KB	16	1	25527			1.542	16554
2	200B – 5KB	16	2	51054			3.124	16343
3	200B – 5KB	16	3	76581			4.706	16273
4	200B – 5KB	16	5	127635			7.801	16361
5	200B – 5KB	16	10	255270			15.712	16247
6	200B – 5KB	16	15	382905			23.493	16299
7	200B – 5KB	16	20	510540			31.254	16335
8	5 – 10KB	6	1	44015			0.6	73358
9	5 – 10KB	6	2	88030			1.251	70368
10	5 – 10KB	6	3	132045			1.822	72473
11	5 – 10KB	6	5	220075			3.064	71826
12	5 – 10KB	6	10	440150			6.158	71476
13	5 – 10KB	6	15	660225			9.223	71585
14	5 – 10KB	6	20	880300			12.347	71297
15	10 – 100KB	7	1	338741	0.31	1092713	1.101	307667
16	10 – 100KB	7	2	677482	0.62	1092713	2.213	306137
17	10 – 100KB	7	3	1016223	0.921	1103391	3.434	295930
18	10 – 100KB	7	5	1693705	1.542	1098382	5.628	300943
19	10 – 100KB	7	10	3387410	3.034	1116483	10.955	308086
20	10 – 100KB	7	15	5081115	4.568	1107962	16.974	299347
21	10 – 100KB	7	20	6774820	6.088	1112815	22.662	298951
22	100KB – 1MB	15	1	4523159	3.434	1317169	7.9	637963
23	100KB – 1MB	15	2	9046318	7.04	1284988	14.19	637514
24	100KB – 1MB	15	3	1.4E+07	10.615	1278330	20.649	657149
25	100KB – 1MB	15	5	2.3E+07	17.595	1285354	34.96	646905
26	100KB – 1MB	15	10	4.5E+07	35.28	1282075	70.14	644876
27	1 – 2MB	5	1	7638636	6.018	1269298	10.715	712892
28	1 – 2MB	5	2	1.5E+07	11.786	1296222	18.516	825085
29	1 – 2MB	5	3	2.3E+07	18.296	1252509	28.2	817841
30	1 – 2MB	5	4	3.1E+07	24.244	1260293	36.662	833412
31	1 – 2MB	5	5	3.8E+07	29.953	1275104	47.47	811809
32	2 – 3MB	5	1	1.3E+07	10.414	1242150	16.003	808333
33	2 – 3MB	5	2	2.6E+07	21.971	1177529	31.765	814466
34	2 – 3MB	5	3	3.9E+07	31.204	1243663	46.106	841496
35	3 – 4MB	5	1	1.8E+07	14.06	1298844	22.241	821085
36	3 – 4MB	5	2	3.7E+07	31.34	1176886	43.662	836505
37	Over 4MB	6	1	3E+07	25.256	1191124	38.645	778446

Figure 8: Test results for the baseline desktop PC





<u>Figure 9: Total time for processing schema validation, signatures, and encryption</u>

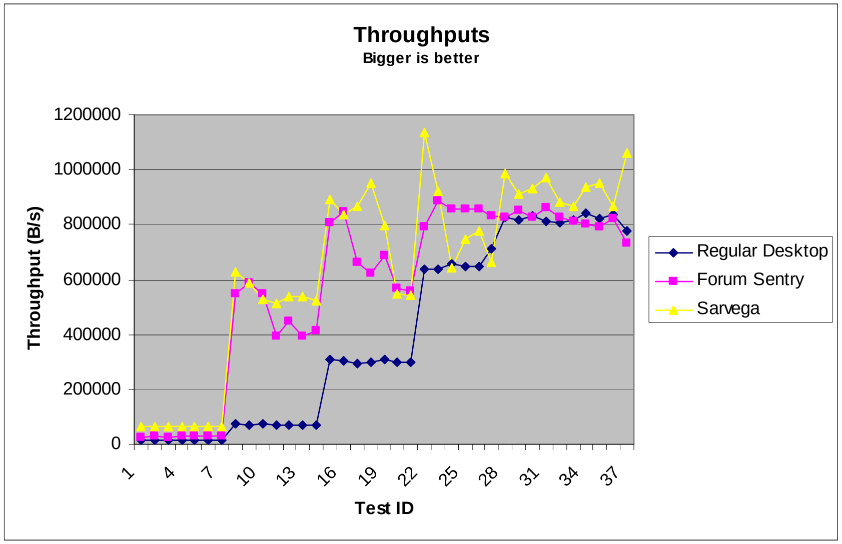
Notes: Most of the time, the Sarvega appliance performed best, but occasionally the Forum appliance very slightly outperformed the Sarvega box. Both are significantly better than the desktop solution as regards overall processing time across a variety of document sizes.

<u>Figure 10: Graph of *x* values for each appliance.</u>

Notes: This shows that the Sarvega appliance incurred significantly less overhead processing as a percentage of its overall processing time than did either the Forum box or the desktop solution. This would be expected for the desktop solution, which incurs significant natural overhead in order to perform XML processing. This initial overhead, though, decreases with document size. It is interesting that all three experienced a significant drop in latency overhead when the individual file sizes increased above 5Kb.

$$x = \frac{[time\_for\_SV \,\&\, Sign \,\&\, Enc] - [time\_for\_SV]}{[time\_for\_SV]} * 100\%$$

$$= \frac{[time\_for\_SV \,\&\, Sign \,\&\, Enc]}{[time\_for\_SV]} * 100\% - 100\%$$



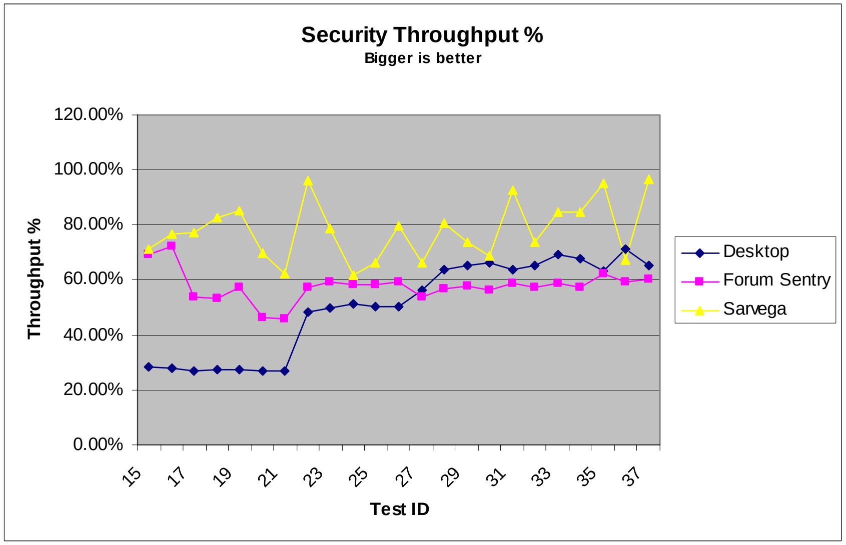


Figure 11: Throughputs for each test

Notes: The Sarvega appliance had a higher overall throughput rate than did either the Forum box or the desktop solution across almost all job sizes and job types (that is, for straight XML processing and for performing security-related tasks). It is interesting that the desktop performance improves significantly relative to the appliances as document size increases, and in some cases actually surpasses that of the Forum appliance.

Figure 12: Graph of *y* values for each appliance

Notes: This demonstrates along with Figure 8 that Sarvega achieves significantly less throughput reduction. Key points on this curve show that Sarvega sometimes achieve near-normal throughput (only slightly less than 100%). At its peak the Sarvega appliance accomplished the security processing with 96% of normal throughput, indicating a <4% hit to throughput.

$$y = \frac{[throughput\_for\_SV \,\&\, Sign \,\&\, Enc]}{[throughput\_for\_SV]} * 100\%$$

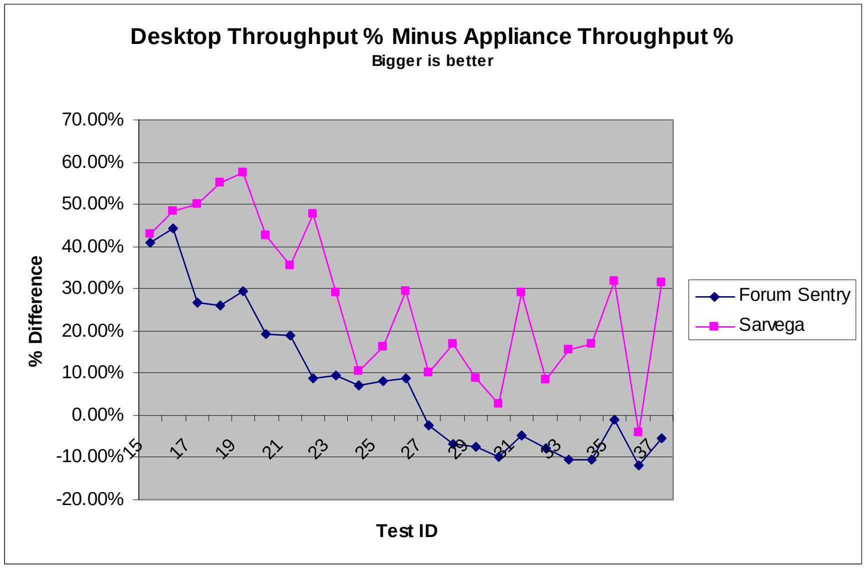


Figure 13: Expected gains of the Sarvega and Forum appliances over the desktop solution

Notes: This graph shows the percentage differences between the Sarvega and Forum appliance performances as compared to our desktop solution. The Sarvega appliance outperformed the desktop solution, in most cases dramatically, in all but one test. The Forum box performed better than the desktop solution in many cases, but as file sizes increased, its performance differential decreased to below the level of the desktop solution.

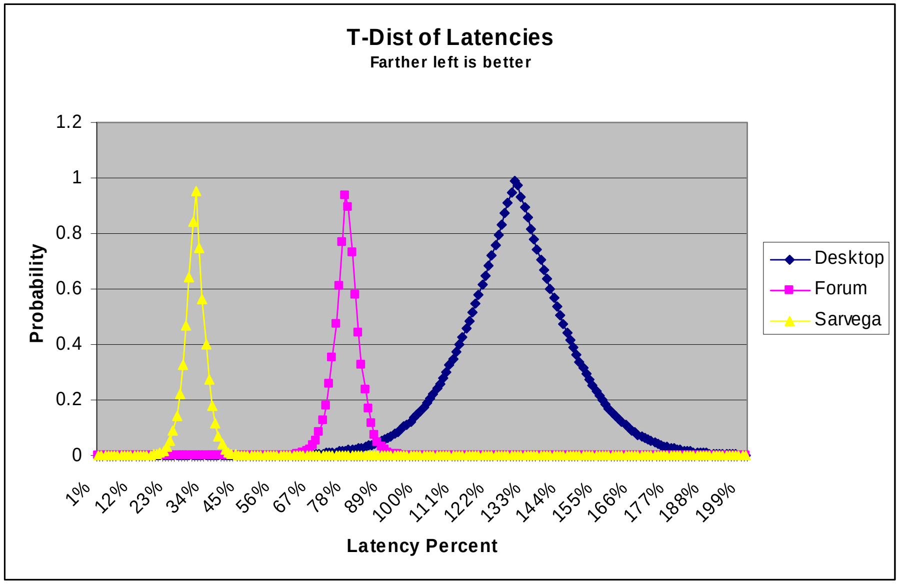


Figure 14: T-distributions of *x* values

Notes: This graph shows the dramatic differences in mean security latencies of each appliance.

See notes for Figure 7 for definition of *x*.

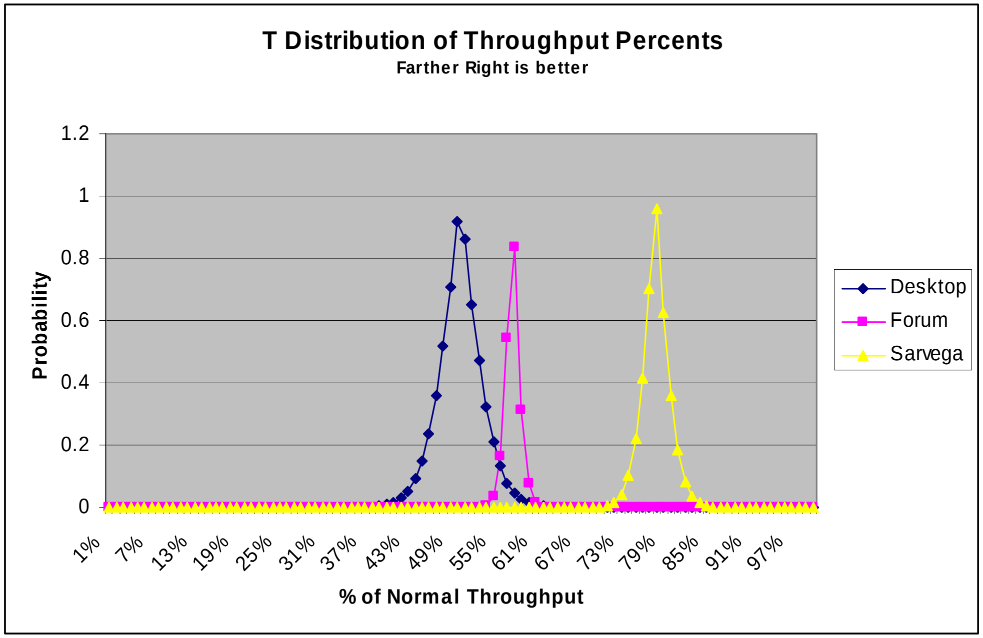


Figure 15: T-distributions of y values

Notes: This graph shows the dramatic differences in mean throughputs for each appliance.

See notes for figure 9 for definition of y.

Appendix B: SUPPLEMENTARY EFFORTS

In addition to the basic testbed interface for timing and comparing performance, we engaged in numerous other studies to investigate the functionality capabilities of using XML and XML appliances in cross-domain use cases. We developed three utility programs to support the experimentation with the XML appliances. We also investigated various methods of dirty-word filtering using XML standards. The results of these related efforts are outlined below

Dispatcher Program: The first is a dispatcher program (see "XML Dispatcher" interface below) that can be used to communicate with and control the flow of data files to the appliances. The dispatcher served as a precursor for our final testbed interface. The functionality of the dispatcher was that it allowed the user to specify the destination web service by IP, port, and extension and it allowed the user to specify the xml file to send, and then number of times to send it. Similar programs are widely available, but we had difficulty finding a single one that provided all of these necessary functionalities in one point.

Figure 16: Screen shot of the Dispatcher

Jabber Client and Server: The second program is actually a client-server pair of programs for testing an XML chat communications protocol, called Jabber, with the appliances. The screen shots below (Figures 15a and 15b) show, respectively, the Jabber server and the Jabber client in action.

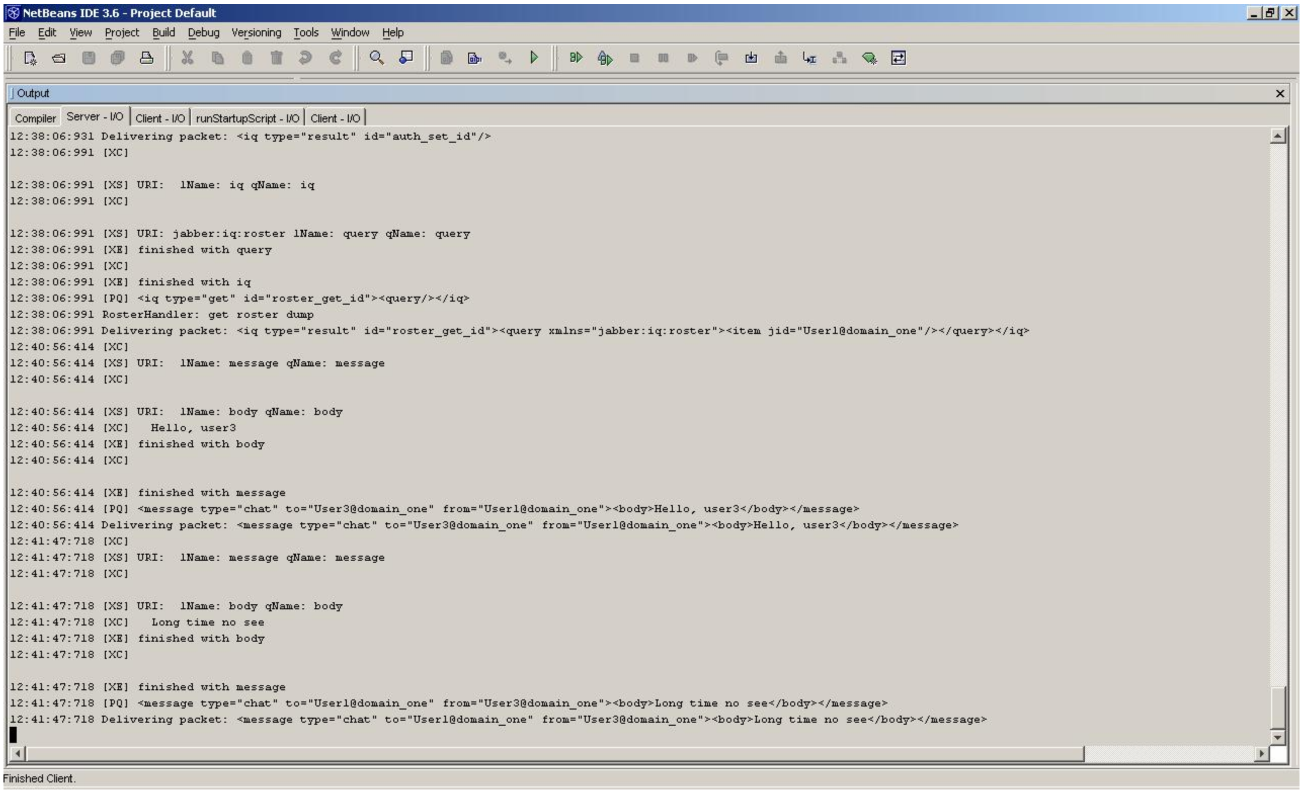


Figure 17a: Jabber server screen shot

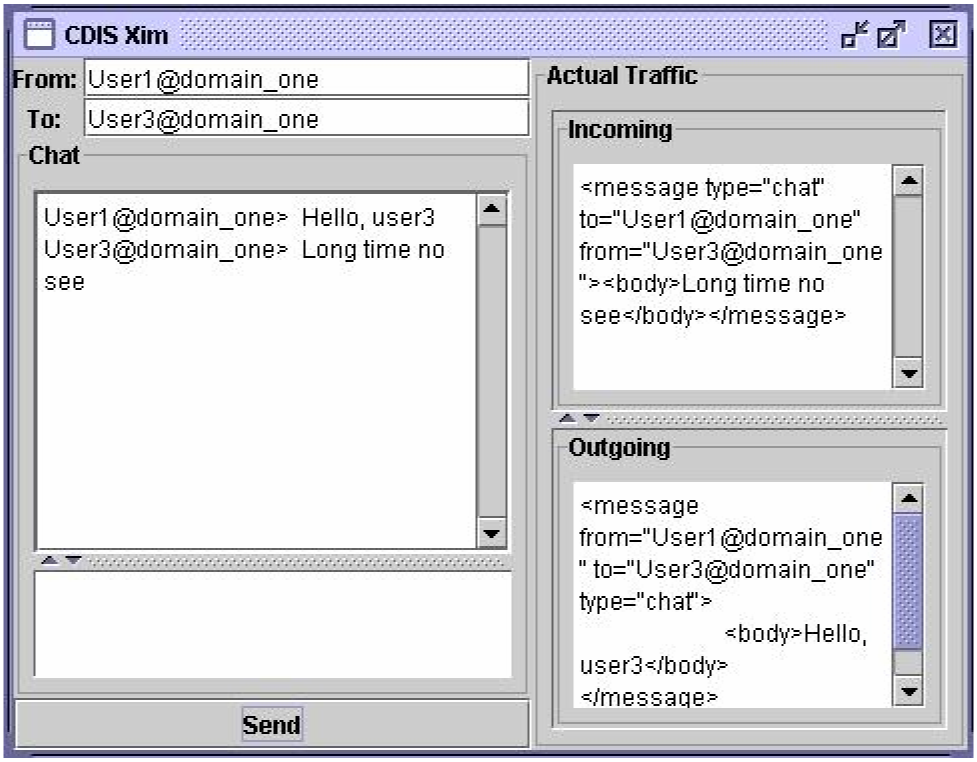


Figure 17b: Jabber client screen shot

Dirty Word Filter generator: Finally, we developed a program to support the production of XSLT stylesheets in a user-friendly way. This was a follow-on effort to the research conducted by one of our intern students, Ryan Cunningham. Ryan discovered several methods of using XML specifications to filter out specific words from portions of an XML document. Of these methods, the most versatile was using XSLT stylesheet transformations. Since stylesheets are the primary means by which document transformation occurs on the XML appliances, and since a canonical use of document transformation in guarding environments is the development and application of dirty word filters, we developed a simple interface (see Figure 16 below) for specifying word replacements to occur as a part of some document processing. The program generates an XSLT stylesheet that incorporates the processing required to do the word replacements, which can easily be incorporated into the processing pipeline of any of the appliances.

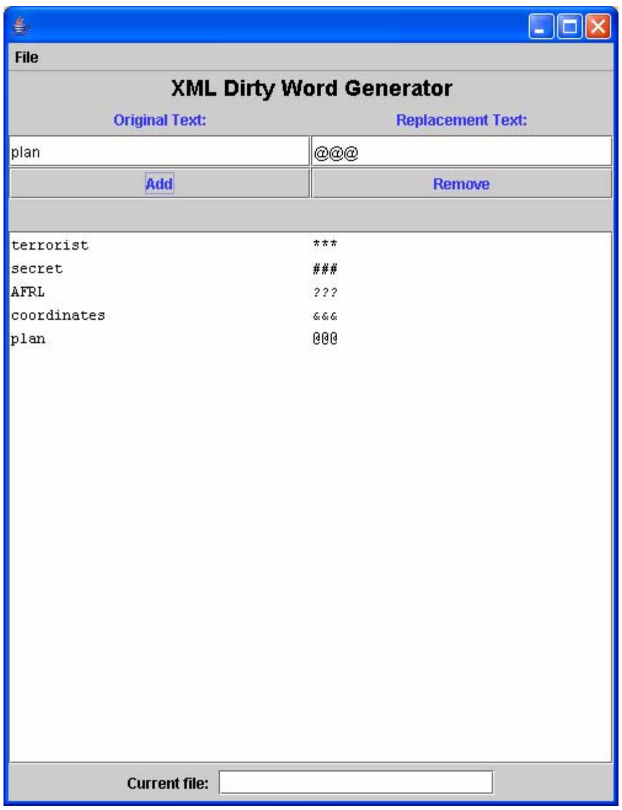


Figure 18: Dirty Word generator screen shot

Genetic Algorithm Research: During the summer of 2005 we had three part-time summer interns working at least part-time with the CDIA S&T group. Each conducted a project of his own that related to their interests and to the long-term research interests of the CDIA S&T group. Of particular interest was Justin Randall's project, in which he investigated the application of genetic algorithms to regular expression development. Regular expressions (generic text-matching patterns) are extremely useful in specifying document filtering and transformation capabilities, but are difficult to define and are often applied to a "moving target" - that is, to documents that are intentionally changed to avoid filtering. We have invited Justin back to continue his work during his winter break from school.

Appendix C: Functionality Comparison Chart

Function Type	Feature		Forum Sentry	Sarvega Guardian
Content Inspection / Modification	Schema Validation	Validate XML messages against a specified XML schema file.	✓	✓
	Regular Expression or Dirty Word Filtering	Filter messages based on the message content, given a regular expression or a specific word.	✓	✓
	XSLT Transformation	Transform XML messages using an specified XSLT stylesheet.	✓	✓
	DoS Protection	Enforce limits to help protect denial of service attack caused by structural XML threats such as coercive parsing and oversized payloads.	✓	✓
	Message Routing	Specify where the received messages should route to.	✓	✓
Message-based Security	Digital Signature	Sign XML messages using the standard XML signature algorithm.	✓	✓
	Signature Validation	Validate signatures from signed XML messages.	✓	✓
	Encryption	Encrypt XML messages using the standard XML encryption algorithm.	✓	✓
	Decryption	Decrypt encrypted XML messages	✓	✓
Identity-based Security	IP Address Filtering	Only accept messages from a set of authorized IP addresses.	✓	✓
	User Authentication	Validate if a user if who he or she claims to be.	✓	✓
	User Authorization	Determine what services that a user is permitted to use.	✓	✓
Performance Enhancement	XML Acceleration	Hardware or software that enhances the performance of processes that handle XML messages.	✓	✓
	SSL Acceleration	Hardware or software that expedites the SSL process.	✓	✓

Operations Management	Logging	Record events that occurred.	✓	✓
	Alerting	Inform an administrator when necessary.	✓	
	Archiving	Backup log files to an external database.	✓	✓
Form Factor	Software	A program that runs on a computer.	✓	
	PCI Card	A piece of hardware that can add onto a computer.	✓	
	Appliance	A stand-alone device that is fully functional.	✓	✓

List of Symbols, Abbreviations, and Acronyms

CCE: Common Criteria Evaluation
CDIA S&T: Cross Domain Information Access, Science & Technology
COTS: Commercial Off-The-Shelf
DoD: Department of Defense
GB: Gigabyte
GHz: Gigahertz
GOTS: Government Off-The-Shelf
IBM: International Business Machines
IIS: Internet Information Services
ISSE Guard: Intelligence Support Server Environment Guard
MTIX: Moving Target Indicator eXploitation
PC: Personal Computer
RAM: Random Access Memory
STAR Guard: Secure Trusted Automated Routing Guard
W3C: World Wide Web Consortium
XML: eXtensible Markup Language
XSLT: eXstensible Stylesheet Language Transformation